In the Claims:

1. (Currently amended) A system for connecting an application to a database having a database management system in order to perform a transaction, said application residing on an application server and said database residing on a database server, said system comprising:

a software driver for connecting said application to said database, wherein a transaction request is made from said application to said database via said software driver;

~~a software library residing on said database server, said software library including at least one procedure for carrying out said transaction request~~;

a distributed transaction coordinator unit residing on said database server, said distributed transaction coordinator being arranged for receiving said ~~database~~ transaction request~~ from said application~~; ~~and~~

a software transaction switch residing on said database server, said transaction switch being arranged for routing said transaction request to the distributed transaction coordinator; and

a software library residing on said database server, said software library including at least one procedure for mapping said transaction request to said transaction switch.

2. (Original) The system of claim 1, wherein said at least one procedure communicates with said transaction switch to start a transaction process in said database based upon said transaction request, and said database receives results of said transaction process and returns results to said distributed transaction coordinator, which returns said results to said application.

3. (Original) The system of claim 1, wherein the distributed transaction coordinator includes a dynamic load library, which provides distributed transaction management control.

4. (Original) The system of claim 1, wherein said software driver is a wire protocol driver that provides direct TCP communications sessions with the database server.

5. (Currently amended) The system of claim ~~2~~4 wherein said wire protocol driver includes interfaces to allow the database server to implement JTA distributed transactions.

6. (Original) The system of claim 1, wherein said software driver is arranged to send request packets to said database to execute said at least one procedure.

7. (Currently amended) The system of claim ~~3~~1, wherein said software transaction switch is arranged to make calls to said distributed transaction coordinator upon receipt of the ~~distributed~~ transaction request ~~from said application~~.

8. (Currently amended) A system for connecting an application to a database having a database management system in order to perform a transaction, said application and said database residing on a database server, said system comprising:

a software driver for connecting said application to said database, wherein a transaction request is made from said application to said database via said software driver;

a distributed transaction coordinator unit residing on said database server, said distributed transaction coordinator being arranged for receiving said ~~database~~ transaction request ~~from said application~~;

~~a software library residing on said database server, said software library including at least one procedure for carrying out said transaction request; and~~

a software transaction switch residing on said database server, said software transaction switch being arranged for routing said transaction request to the distributed transaction coordinator; and

a software library residing on said database server, said software library including at least one procedure for mapping said transaction request to said transaction switch.

9. (Currently amended) A method for performing a distributed transaction by connecting an application to a database having a database management system, said application residing on an application server and said database residing on a database server, comprising:

making a transaction request, by sending a request packet from said application server to said database to execute a plurality of stored procedures on said database;

mapping said transaction request to a transaction switch residing on said database server using at least one of said stored procedures;

sending said transaction request, via said transaction switch, to a library in a distributed transaction coordinator residing on said database server~~.~~;

returning a status of said transaction request to the application server; and

reporting said status of said transaction request to the application.

10. (Original) The method of claim 9, further comprising: storing said stored procedures in a library residing on said database.

11. (Currently amended) The method of claim 9, wherein ~~making a transaction request and~~ sending a request packet includes: ~~making the transaction request; and~~ sending ~~a~~ said request packet via a wire protocol driver.

12. (Original) The method of claim 11, further comprising: including interfaces in said wire protocol for implementing JTA distributed transactions.

13. (Currently amended) The method of claim 11, further comprising: allowing a transaction to enlist a distributed transaction with at least one of said ~~plurality of~~stored procedures.

14. (Currently amended) The method of claim 9, ~~further comprising: said plurality of procedures~~ wherein mapping said transaction request to said transaction switch includes making calls into a switch library ~~unit~~ residing on said database server to connect the application to the database.

15. (Currently amended) A method for performing a distributed transaction by connecting an application to a database having a database management system, said application residing on an application server and said database residing on a database server, comprising:

transmitting a start request transmitted from said application to said database;

calling an extended stored procedure from a plurality of stored procedures stored in said database based on ~~a~~ said start request;

calling, by the extended stored procedure, a start function in a transaction switch library residing on said database server;

obtaining, by the extended stored procedure, a transaction ID from said database and returning said transaction ID to a said application server; and

sending a data packet to said database along with the transaction ID instructing the database server to enlist a specified transaction in a distributed transaction.

16. (Original) The method of claim 15, wherein returning said transaction ID includes: returning the transaction ID in the form of a transaction cookie.

17. (Original) The method of claim 15 wherein transmitting said start request includes: transmitting the start request via a driver on the application server.

18. (Original) The method of claim 15, further comprising: storing said plurality of stored procedures in a library located on said database.

19. (Original) The method of claim 15, wherein sending a data packet includes: sending said data packet via a wire protocol driver.

20. (Original) The method of claim 19 further comprising: including interfaces in said wire protocol driver for connecting to said database.

21. (Original) The method of claim 15, further comprising: allowing a transaction to enlist a distributed transaction with one of said plurality of procedures.

22. (Currently amended) The method of claim 15, further including: at least one of said plurality of procedures making calls into ~~a~~ said transaction switch library ~~unit~~ to connect the application to the database.

23. (Currently amended) The method of claim 15, further including: said transaction switch library making calls to ~~said~~ a distributed transaction coordinator.